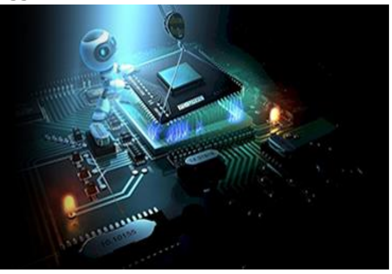
International Journal of Engineering in Computer Science 2024; 6(2): 51-55

International Journal of Engineering in Computer Science



E-ISSN: 2663-3590
P-ISSN: 2663-3582
www.computersciencejournals.com/ijecs
IJECS 2024; 6(2): 51-55
Received: 12-06-2024
Accepted: 18-07-2024

Dr. M Vanitha
Professor, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad, Telangana, India

Batta Mayuri
Student, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad, Telangana, India

Bollu Radhika
Student, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad, Telangana, India

Basi Navya
Student, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad, Telangana, India

Corresponding Author:
Dr. M Vanitha
Professor, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad, Telangana, India

Cloud computing data centre: SDN controller network balancing load method

Dr. M Vanitha, Batta Mayuri, Bollu Radhika and Basi Navya

DOI: https://doi.org/10.33545/26633582.2024.v6.i2a.125

Abstract
A load balancing strategy for software defined networking (SDN) in the cloud seems to resolve the availability and latency difficulties in the HTTP server nowadays, thanks to the strong demand for cloud services. In this article, we show how to implement a load balancing system for software defined networking (SDN) in the cloud. The HTTP server handled 6 million requests, with 400 requests per second. Due to overload, the website's server will not reply to any more requests sent by the client. hence, after applying load balance, the same parameters may be measured using a load balancing system. The 6 million requests, each involving four connections, were all successfully accepted by the cloud environment's HA proxy, allowing the traffic from the HTTP server to be dispersed across a pool of servers rather than a single server. Having an HTTP load balancer helps to improve the availability and latency of HTTP requests, and the HA proxy eventually gives an adequate answer time with no requests lost, even after 6 million requests arriving at the same time.

Keywords: Cloud computing, data centre, software defined networking (SDN)

Introduction

Particularly in the realm of computer networks, the creation of new technology is very beneficial. The ease with which users can access their data and other resources is greatly enhanced by its crucial function in facilitating information exchange throughout the network. Researchers now have a greater responsibility to guarantee the security and high availability of information and communication technology infrastructures in the cloud, since cloud computing has become one of the most prominent global technological developments due to these needs. "Traditional Network" has been the name given to the older network architecture since the inception of software-defined networking (SDN) ideas, while "Modern Networks" has been used to describe SDN networks. According to the Open Systems Interconnection (OSI) paradigm, the switching device in Layer 2 and the router in Layer 3 are responsible for controlling the data plane and control plane, respectively, in conventional networks. Two common paradigms in traditional networks are client-server and peer-to-peer. The idea behind software-defined networking (SDN) is to create a new device, the controller, to manage traffic after separating the control layer from the information plane. Part A: Defined Networking via Software With the use of a centralized control interface termed a "controller," software-defined networking (SDN) aims to let cloud computing experts, network technicians, and system administrators swiftly adapt to changes in business needs.
Controlling network traffic is the exclusive responsibility of the controller. A load balancer may divide the workload across servers, controls, or any other resource, meeting the need for high availability. There is a dizzying array of controller kinds, each tailored to a certain function and language of development. Several factors, such as the language it was written in, as well as others, such as throughput, latency, bandwidth, buffer size, or resource consumption, may be used to determine the sort of controller to choose. As a result, software-defined networks may make better use of programmable devices like load balancers, traffic engineers, and others, all via the network manager's interface, without requiring direct alterations to physical devices. A new kind of network management software called Next-Generation Network (NGN) was unveiled. Creates a physical separation between the control layer and the data plane by directing their management to a separate computer called the controller. B. System for Balancing Loads In order to prevent a complete

failure of any one of the aforementioned nodes, load balancing distributes the workload over all of the nodes in the network, including computers, controllers, routers, and switches. In addition to being implemented at each level of the OSI model, the load balancing method may also be accomplished by specific hardware or software. The most basic kind of load distribution is DNS load balancing, which is in charge of translating IP addresses into domain names. In this case, the load balancing is straightforward because each server uses a pool of Internet Protocol (IPS) addresses; in the event that one of these addresses fails, the other will take over. However, this approach has the drawback of not being able to tell which machines are offline. Distributing servers across many geographic regions ensures availability in the event of a natural catastrophe, which is the most crucial consideration. A round robin mechanism will be used to allocate the IP addresses. Our server's Internet Protocol (IP) address is initially allocated to the very first domain name that is requested. After that, the second domain name's IP address, and so on. When one server's IP cycle reaches 'n,' the process restarts from the beginning.

Related work

A Software-Defined Network with Load Balancing,
P. Kumari and D. Thakur, 2017 [11]

The term "software defined networking" (SDN) refers to a network architecture that allows for lower-level functional abstraction and open interfaces to let administrators establish, oversight, control, and dynamically alter the network's behavior. Software-defined networking (SDN) is based on the premise that conventional networks' static design isn't up to the task of supporting the storage needs, scalability needs, and dynamic nature of contemporary computing. This is made feasible by the use of dissociation or separation procedures inside the system, which aid in decision-making about the delivery of traffic from the systems and subsequently conveys this traffic to its intended destination. In a computer network, load balancing is a technique for dividing up tasks across several computers so that they all work together to finish the job within the allotted time. Depending on the situation, load balancing may be implemented using either software or hardware. Load balancing also leads to clustering of computer servers. In order to better use resources and provide linear service among several customers in a software-defined networking (SDN) environment, this article explores the different types of load balancing algorithms.

"A Information Layer Multi-Path balancing of load Method for Hybrid Networks Based on Software in Different Topologies,"
Themba Shozi, Praga Sen Mudale, Matthew O. Adigun, Olabisi Kobo 2019 [12]

While software-defined networking (SDN) offers great promise for improving networking, it also raises additional economic and commercial concerns, such as limited resources, lack of consensus among network operators, and uncertainty about the future. Consider a hybrid SDN deployment as a potential solution. It combines the strengths of both SDN and traditional system capabilities, allowing organizations to adapt to budget constraints, utilize time-tested traditional tools, and achieve a seamless transition to SDN. The need to achieve increased network performance, minimize delay, and decrease the time required for network response has arisen due to the fact that modern-day communication is now an integral part of day-to-day corporate processes. A hybrid software-defined network (SDN) data plane with multi-path load balancing is designed and studied in this research. Four distinct hybrid SDN topologies—the Fat Tree, Ring, Mesh, and Torus topologies—are used to evaluate the load balancer's performance with the Free Day Light controller. The primary outcome of this study is an analysis and testing framework for data plane congested and link overload load balancing, which may be used for deployment in hybrid SDN systems either before or after a network breakdown. The results show that the network's overall performance and latency have been enhanced.

"A cloud computing load-balancing technique inspired by artificial bee colonies,"
Ullah, Arif. 2019 [22]

Information technology is seeing the rise of cloud computing. Problems such as load balancing persist, however. This method equitably distributes work across several nodes in a dynamic environment where certain nodes are underloaded and others are overloaded. Load balancing is mostly successful in reducing energy use and resource utilization. When traditional, mathematical approaches fail to give satisfactory results, swarm intelligence steps in to fill the void. Parabola created an algorithm in 2005 that mimics the foraging behavior of artificial bee colonies. It is very adaptable, strong, and converges quickly. A separate group of researchers improved load balancing using the ABC method. The ABC algorithm for load balancing in the cloud is the subject of this in-depth research study. Additionally, it provides definitions of key terms related to swarm intelligence and its characteristics.

"An SDN-based IP jumping messaging scheme versus scanning attack,"
Fei Xiang, Yan Li. 2017 [14]

The safety of data transmissions via networks is greatly jeopardized by scanning attacks. Scanning attackers find the use of static IP address setting in traditional network communication quite convenient. In order to overcome this issue, this article proposes an IP Jumping (IPH) communication strategy that is based on software-defined networks. This method allows for the dynamic assignment of IP addresses to hosts in the network, enabling dynamic network communication. IPH allows hosts to have their IP addresses changed in real time without disrupting ongoing connections; it is a live internet address allocation mechanism. Furthermore, IPH can also handle multi-channel protocols. The results of the experiments demonstrate that IPH is capable of supporting multi-channel protocols at a reasonable cost and effectively defending against scanning assaults.

"Chapter 30 The Protocol for Address Resolution Based Attacks: Protection and Monitoring Schemes,"
D. Francis Xavier Christopher, C. Divya. 2020 [15]

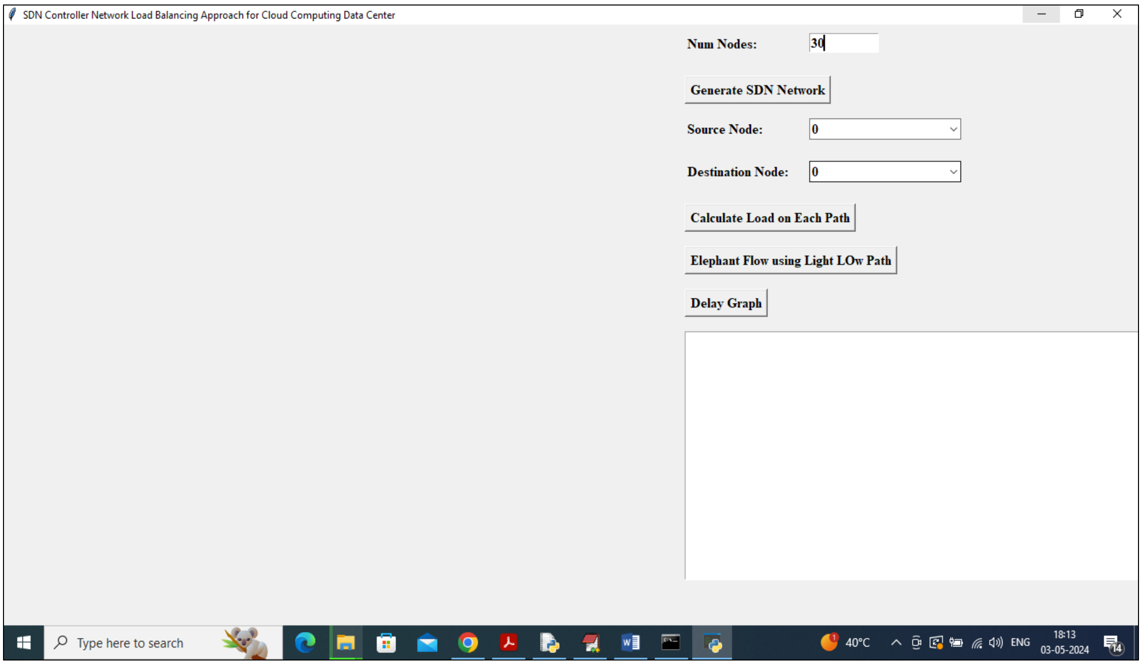
Because of its widespread use at both the individual and corporate levels, resolution of issues protocol (ARP) has earned a reputation as a crucial protocol in the field of computer communications. Although effective in most situations, ARP was not intended to deal with hostile hosts.

Given these challenges, several academics have worked on ways to make ARP more successful in detecting and blocking malicious attacks like as spoofing, DoS, and MIST (Man in the middle). Reviewing and analyzing the many strategies created for identifying attacks and mitigation, this study concentrates on the ARP poisoning assaults and their benefits. In addition to comparing various attack mitigation and detection approaches based on important characteristics, they are also appraised. In order to better identify and mitigate ARP attacks, this research sheds light on the current method and provides a foundation for future improvements.

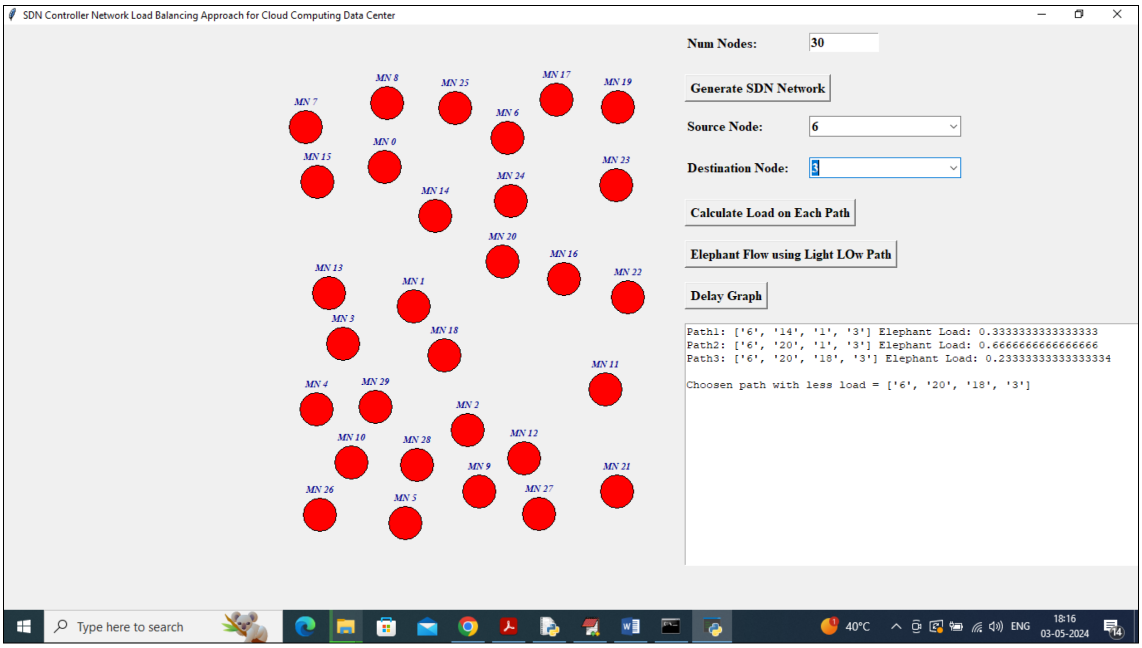
Methodology

1) The following components make up this project, which we have executed as a simulation.
2) Create Software-Defined Network: This module will be used to create a simulated network with nodes randomly positioned.
3) Determine Load on Each Path: This module allows users to identify the source and destination nodes. The Elephant SDN scheduling method then finds all possible pathways between the two, calculates the available load, and chooses the path with the least load.
4) Fourthly, the Elephant Flow with Light Low Path allows packets to go from source to destination via paths with little burden. It will determine how much time has passed from the moment of transmission for each packet until it reaches its final destination.
5) latency Graph: will create a graph that compares the proposed method to the current state of the art in terms of latency.

Results and Discussion

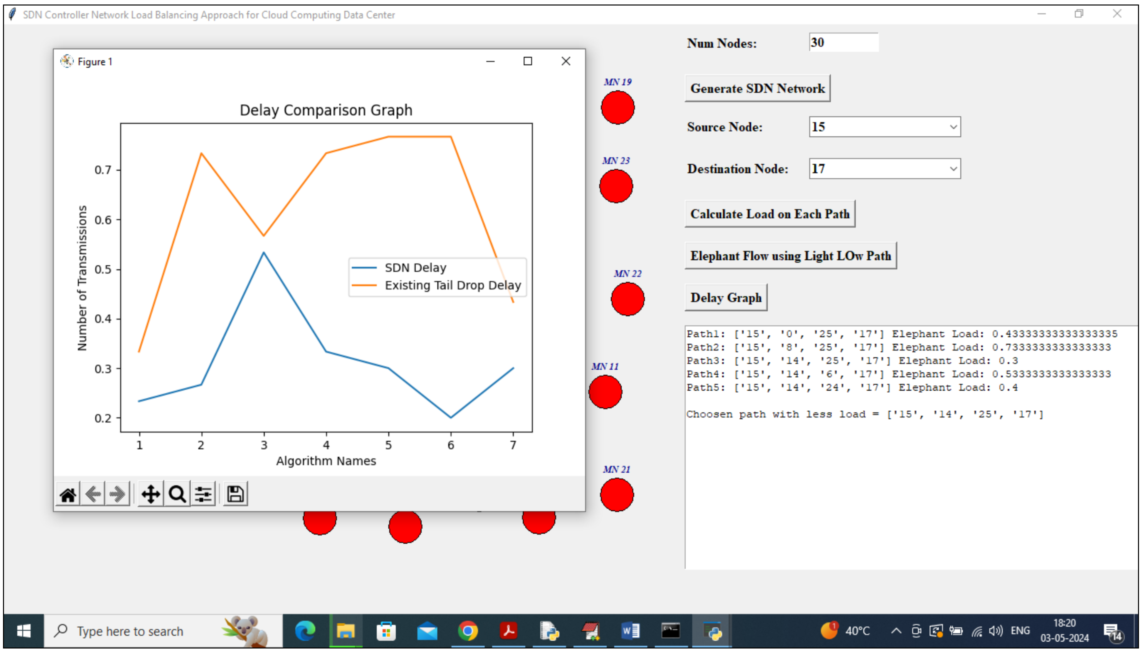


After entering 30 as the "Number of Nodes" in the previous result, I clicked the "Generate SDN Network" button, which brought me the screen below.



Below the text section of the above result, you can see three pathways that have been discovered from the source to the destination. Next, you can see the available load for each path. Finally, you can see the one that has the least load, and you may click on it.

On 'Elephant Flow using Light Low Path' button to send data from source to destination using above path



The number of packets transmitted from the source to the destination is represented by the x-axis of the delay graph, which also shows the current "Tail Drop Technique" and the proposed "SDN" elephant technique; the blue line shows the latter, which has a lower delay than the orange line.

Conclusion

The results show that Hap Roxy maintains a good response time and does not drop any requests even when faced with 6 million requests all at once. In this case, the Round Robin algorithm works well for both Client 1 and Client 2. If your pools are not identical, you should use weighted round robin, which gives more weight to pools with more resources and less to those with less. It works well in this scenario. The traffic from the HTTP server doesn't have to go via a single server; in fact, it may be spread out across a pool of servers. In addition, both availability and latency were seen to have improved after the implementation of the load balancer mechanism in the HTTP server. Installing and configuring the OpenStack cloud might be a real pain in the end. Prior knowledge of the OpenStack cloud structure, installation goals (single node vs. distributed system, where each computer handles one or more modules), and Linux distribution to be used for installation should be carefully considered.

References

1. Armbrust M, Fox A, Gruber R, *et al*. A view of cloud computing. Commun ACM. 2010;53(4):50-58.
2. McKeown N. Software-defined networking. INFOCOM keynote talk. 2009;17(2):30-32.
3. Wette Roth D. OSI reference model for telecommunications. New York: McGraw-Hill; c2002. Vol. 396.
4. Ranjani S. Dynamic load balancing using Software Defined Networks.
5. Sezer S, Nikravesh M, Rexford J, *et al*. Are we ready for SDN? Implementation challenges for software-defined networks. IEEE Commun Mag. 2013;51(7):36-43.
6. Zakia U, Yader HB. Dynamic load balancing in SDN-based data centre networks. In: 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE; c2017.
7. Subramanian S, Voruganti S. Software-Defined Networking (SDN) with OpenStack. Livery Place: Packet Publishing Ltd.; c2016.
8. Briscoe N. Understanding the OSI 7-layer model. PC Network Advisor. 2000;120(2).
9. Senthil Ganesh N, Ranjani S. Dynamic load balancing using software defined networks. Int J Comput Appl; c2015.
10. Yousif YA. SDN Controller performance analysis in Datacentre. Future University Journal; c2018.
11. Kumari P, Thakur D. Load balancing in software defined network; c2017.
12. Shozi T, Mudale PS, Adigun MO, Kobo O. A data plane multi-path load balancing mechanism for hybrid software defined networks in different topologies. In: 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (ICED); c2019.
13. Zhao Z, Liu F, Gong D, Chen L, Xiang F, Li Y. An SDN-based IP hopping communication scheme against scanning attack. In: 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN); c2017.
14. Xiang F, Li Y. An SDN-based IP hopping communication scheme against scanning attack. In: 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN); c2017.
15. Christopher DFX, Divya C. Chapter 30 Address Resolution Protocol based attacks: Prevention and

detection schemes. In: Springer Science and Business Media LLC; c2020.
16. Nadeau TD, Gray K. SDN: Software Defined Networks: an authoritative review of network programmability technologies. O'Reilly Media, Inc.; c2013.
17. Calin D, Nam HW. Optimizing quality of service in a content distribution network using software defined networking. US Patent No. 10,028,167; c2018.
18. Zaw HT, Maw AH. Traffic management with elephant flow detection in software defined networks (SDN). Int J Electr Comput Eng. 2019;9(4):3203-3211. DOI: 10.11591/piece.v9i4.pp3203-3211.
19. Liyanage M, Gurov A, Ylianttila M. Software defined networking concepts. In: Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture. John Wiley & Sons, Ltd; c2015. p. 21-44.
20. Jain R, Paul S. Network virtualization and software defined networking for cloud computing: A survey. IEEE Commun Mag. 2013;51(11):24-31.
21. Kreutz D, Ramos FMV, Veríssimo PE, Rothenberg CE, Anodally S, Uhlig S. Software-Defined Networking: A comprehensive survey. Proc IEEE. 2015;103(1):14-76.
22. Ullah A. Artificial bee colony algorithm used for load balancing in cloud computing. IAES Int J Artif Intell. 2019;8(2):156.
23. Umar S, Baseer S. Perception of cloud computing in universities of Peshawar, Pakistan. In: 2016 Sixth International Conference on Innovative Computing Technology (INTECH); c2016. p. 87-91.
24. Ullah A, Nawi NM, Shahzad A, Khan SN, Aamir M. An e-learning system in Malaysia based on green computing and energy level. JOIV: Int J Informatics Visual. 2017;1(4-2):184-187.
25. Devi D, Chitra V, Rhymed Thurairajah V. Load balancing in cloud computing environment using improved weighted round robin algorithm for nonredemptive dependent tasks. Sci World J. 2016;2016.
26. Pradhan P, Behera PK, Ray BNB. Modified round robin algorithm for resource allocation in cloud computing. Procedia Comput Sci. 2016;85:878-890.
27. Mahajan K, Makro A, Dahiya D. Round robin with server affinity: a VM load balancing algorithm for cloud-based infrastructure; c2013.